

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR LETTERS PATENT

**System and Method to Display and Manage Data within  
Hierarchies and Polyarchies of Information**

Inventor(s):

George G. Robertson

Kim Cameron

Mary Czerwinski

Daniel C. Robbins

ATTORNEY'S DOCKET NO. MS1-623US

1 **RELATED APPLICATIONS**

2 This application claims the benefit of U.S. Provisional Application No.  
3 60/219,796, filed July 19, 2000, entitled "Hierarchy Visualization", to George G.  
4 Robertson and Kim Cameron.

5  
6 **TECHNICAL FIELD**

7 The description relates generally to a graphical user interface of a  
8 programmed computer system. More specifically, the description relates to a  
9 system and procedure for displaying and managing data within multiple  
10 intersecting hierarchies of data.

11  
12 **BACKGROUND**

13 A variety of application programs and operating systems include a user  
14 interface feature enabling users to view a listing of information stored on a  
15 computer or one or more data bases connected to the computer. For example, the  
16 "Windows"-brand operating system from Microsoft Corporation includes a File  
17 Manager program to help a user keep track of files stored in a computer.

18 A hierarchy, or tree structure, is a set of nodes or elements that are  
19 connected to each other with each node except the root node having exactly one  
20 parent. The root node has no parent. Each node has any number of children, from  
21 none to many. A node with no children is called a leaf node.

22 One example of this tree structure is a file system where related information  
23 is grouped together using a hierarchy of subdirectories. In this example, the main  
24 directory is the root node of the tree and any subdirectories to the main directory  
25 are its child nodes. Each subdirectory can be broken into a number of other

1 subdirectories to achieve additional layers of the file system hierarchy. At each  
2 level of the file system hierarchy, there may be a number of files, which are leaf  
3 nodes of the tree structure. In addition to directories of files, the concept of a tree  
4 structure can be extended to other arbitrary groupings of data.

5 While some systems display or manage information in a hierarchical  
6 fashion, recent developments in information technology provide network  
7 administrators with data that are not able to be displayed using such traditional  
8 systems.

9 For example, network administrators trying to maintain disparate  
10 directories of enterprise users and resources are beginning to use metadirectories.  
11 A metadirectory ties disparate directories – such as those associated with  
12 databases, e-mail applications, human resource systems, network operating system  
13 directories, and other corporate systems, into a single, logical directory. The  
14 ability to merge databases and information like this is recent. Traditional systems  
15 and procedures do not provide means to access, display, and manage the complex  
16 relationships between the data provided by a metadirectory. Data about people  
17 and resources is just one example of data. Anytime an enterprise merges various  
18 databases together, the enterprise will typically encounter multiple intersecting  
19 hierarchies of data.

20 There is a need for an improved user interface that enables a user to view  
21 and manage the complex relationships among data presented by metadirectories.  
22  
23  
24  
25

## SUMMARY

The system and procedures described below display and manage the complex relationships between data provided by a metadirectory. One described procedure includes specifying an object having first and second intersecting hierarchies. The object is illustrated with respect to the first hierarchy. The second intersecting hierarchy is presented for selection to illustrate the object with respect to the second intersecting hierarchy. In response to selecting the second intersecting hierarchy, the object is illustrated within the second intersecting hierarchy.

To illustrate the object within the second intersecting hierarchy, the procedure visually morphs from the first intersecting hierarchy to the second intersecting hierarchy. When morphing, the first structure representing the first hierarchy is gradually changed to a second structure that represents the second hierarchy.

In one implementation, the morphing process employs a visual pivot in which the first hierarchical structure is changed to a second hierarchical structure by graphically pivoting about an object that is common to both structures. This visual pivot illustrates the relationship of the objects with respect to the first and second intersecting hierarchies.

## BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram of a system for displaying and managing objects within multiple hierarchies of information.

Fig. 2 is a block diagram of a computer for displaying and managing objects within multiple hierarchies of information.

1 Fig. 3 illustrates a graphical user interface that is capable of presenting  
2 multiple objects within multiple different hierarchies of information. In Fig. 3, a  
3 particular object is presented within a first hierarchy.

4 Fig. 4 illustrates the graphical user interface of Fig. 3, but shows the  
5 particular object within a second hierarchy.

6 Fig. 5 illustrates the graphical user interface of Fig. 3. Fig. 5 shows the  
7 particular object within a first hierarchy and second hierarchy as the first hierarchy  
8 is being morphed to the second hierarchy in a first stage of a visual pivot  
9 technique.

10 Fig. 6 illustrates the graphical user interface of Fig. 3. Fig. 6 shows the  
11 particular object within a first hierarchy and second hierarchy as the first hierarchy  
12 is being morphed to the second hierarchy in a second stage of a visual pivot  
13 technique.

14 Fig. 7 illustrates the graphical user interface of Fig. 3. Fig. 7 shows the  
15 particular object within a first hierarchy and second hierarchy as the first hierarchy  
16 is being morphed to the second hierarchy in a third stage of a visual pivot  
17 technique.

18 Fig. 8 illustrates the graphical user interface of Fig. 3. Fig. 8 shows the  
19 particular object within a first hierarchy and second hierarchy as the first hierarchy  
20 is being morphed to the second hierarchy in a first stage of a visual slide  
21 technique.

22 Fig. 9 illustrates the graphical user interface of Fig. 3. Fig. 9 shows that the  
23 graphical user interface further comprises a search results area to display one or  
24 more objects that correspond to the particular object.  
25

1 Fig. 10 illustrates the graphical user interface of Fig. 9. Fig. 10 illustrates  
2 that upon selection of a different object identifier from the search results area, the  
3 selected additional object replaces any details of the particular object with respect  
4 to the current hierarchy.

5 Fig. 11 illustrates the graphical user interface of Fig. 9. Fig. 11 shows the  
6 selection of objects from the search area and display of multiple objects with  
7 respect to a hierarchy of information.

8 Fig. 12 illustrates the graphical user interface of Fig. 9. Fig. 12 shows a  
9 filter selection area to filter the object identifiers displayed in the search area. Fig.  
10 12 also shows context sensitive information corresponding to displayed objects.

11 Fig. 13 is a flowchart that illustrates a procedure for displaying objects  
12 within multiple hierarchies of information.

Sub A ✓

1 DETAILED DESCRIPTION

2 The following description provides a system and procedure to display and  
3 manage data objects within complex hierarchies and polyarchies of data. In a  
4 hierarchy, data that is arranged in a graded series with respect to other data. A  
5 data polyarchy is two or more intersecting hierarchies of data. In other words, in a  
6 data polyarchy one or more data nodes in a first hierarchy are shared in various  
7 dimensions with one or more other hierarchies of data.

8 To illustrate a polyarchy, or multiple intersecting hierarchies of  
9 information, consider the following example. In a company, a person has both  
10 implicit and explicit relationships not only to other people, but also to company  
11 resources, geographical locations, business units, club memberships, hobbies, and  
12 the like. Each of these relationships respectively represents a hierarchy of data, or  
13 information. Thus, one hierarchy might show an individual within an  
14 organizational chart of people and another hierarchy might depict the individual's  
15 physical location on the company campus. A polyarchy is formed from the  
16 multiple hierarchies because each hierarchy shares a common node along various  
17 dimensions that corresponds to the person.

18 Polyarchies, or multiple intersecting hierarchies of information, become  
19 very complex when they exist with respect to multiple people and/or multiple  
20 resources. To illustrate this, consider that one or more of the above listed  
21 hierarchies may intersect with other respective hierarchies of information that  
22 correspond to other people who work for the same company, live in the same  
23 geographical area, belong to the same clubs, share supervisors, computing  
24 resources, titles, building locations, e-mail lists membership, and the like.

Unfortunately, traditional systems and procedures do not provide means to display and manage the complex relationships between such data.

### **Exemplary System**

Fig. 1 is a block diagram of a system 100 to display and manage data within multiple intersecting hierarchies of information. The system includes a computer 102 that is logically connected across communication pathways 114 to metadirectory services server 106, which in turn is logically connected to one or more databases 110. The logical connections in system 100 can be a local area network (LAN) and a wide area network (WAN). Such networking environments are commonplace in offices, enterprise wide computer networks, intranets, and the Internet.

Databases 110 are object-oriented databases, wherein objects are linked to other objects. For example, Extensible Markup Language (XML) databases and Hypertext databases are examples of well known databases where objects are linked to other objects. The databases represent disparate sources of information – such as information associated with e-mail applications, human resource systems, network operating system directories, other corporate systems, and the like. Databases 110 include can include, for example, SQL server databases.

Metadirectory services server 106 includes a processor (not shown) and operating system (not shown) used to execute a metadirectory services application 108. The metadirectory services application ties the disparate directories of information provided by databases 110 into a single, logical directory. Server 106 serves data from databases 110 to computer 102 in response to queries.



Computer 102 includes a processor (not shown) and an operating system (not shown) that support a graphical user interface 110. Graphical user interface 110 displays data served from metadirectory services server 106 within hierarchies and polyarchies of information.

### **Exemplary Computer**

Fig. 2. shows a computer 102 that forms a suitable environment for the system described above. The components shown in Fig. 2 are only examples, and are not intended to suggest any limitation as to the scope of the functionality of the invention; the invention is not necessarily dependent on the features shown in Fig. 2.

Generally, various different general purpose or special purpose computing system configurations can be used. Examples of well known computing systems, environments, and/or configurations that may be suitable for use with the invention include, but are not limited to, personal computers, server computers, hand-held or laptop devices, multiprocessor systems, microprocessor-based systems, set top boxes, programmable consumer electronics, network PCs, minicomputers, mainframe computers, distributed computing environments that include any of the above systems or devices, and the like.

The functionality of the computers is embodied in many cases by computer-executable instructions, such as program modules, that are executed by the computers. Generally, program modules include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types. Tasks might also be performed by remote processing devices that are linked through a communications network. In a

distributed computing environment, program modules may be located in both local and remote computer storage media.

The instructions and/or program modules are stored at different times in the various computer-readable media that are either part of the computer or that can be read by the computer. Programs are typically distributed, for example, on floppy disks, CD-ROMs, DVD, or some form of communication media such as a modulated signal. From there, they are installed or loaded into the secondary memory of a computer. At execution, they are loaded at least partially into the computer's primary electronic memory. The invention described herein includes these and other various types of computer-readable media when such media contain instructions programs, and/or modules for implementing the steps described below in conjunction with a microprocessor or other data processors. The invention also includes the computer itself when programmed according to the methods and techniques described below.

For purposes of illustration, programs and other executable program components such as the operating system are illustrated herein as discrete blocks, although it is recognized that such programs and components reside at various times in different storage components of the computer, and are executed by the data processor(s) of the computer.

With reference to Fig. 2, the components of computer 102 may include, but are not limited to, a processing unit 120, a system memory 130, and a system bus 121 that couples various system components including the system memory to the processing unit 120. The system bus 121 may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures. By way of example, and not

1 limitation, such architectures include Industry Standard Architecture (ISA) bus,  
2 Micro Channel Architecture (MCA) bus, Enhanced ISA (EISA) bus, Video  
3 Electronics Standards Association (VESA) local bus, and Peripheral Component  
4 Interconnect (PCI) bus also known as the Mezzanine bus.

5 Computer 102 typically includes a variety of computer-readable media.  
6 Computer-readable media can be any available media that can be accessed by  
7 computer 102 and includes both volatile and nonvolatile media, removable and  
8 non-removable media. By way of example, and not limitation, computer-readable  
9 media may comprise computer storage media and communication media.  
10 Computer storage media includes volatile and nonvolatile, removable and non-  
11 removable media implemented in any method or technology for storage of  
12 information such as computer-readable instructions, data structures, program  
13 modules, or other data. Computer storage media includes, but is not limited to,  
14 RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM,  
15 digital versatile disks (DVD) or other optical disk storage, magnetic cassettes,  
16 magnetic tape, magnetic disk storage or other magnetic storage devices, or any  
17 other medium which can be used to store the desired information and which can be  
18 accessed by computer 110. Communication media typically embodies computer-  
19 readable instructions, data structures, program modules or other data in a  
20 modulated data signal such as a carrier wave or other transport mechanism and  
21 includes any information delivery media. The term "modulated data signal"  
22 means a signal that has one or more of its characteristics set or changed in such a  
23 manner as to encode information in the signal. By way of example, and not  
24 limitation, communication media includes wired media such as a wired network or  
25 direct-wired connection and wireless media such as acoustic, RF, infrared and

other wireless media. Combinations of any of the above should also be included within the scope of computer readable media.

The system memory 130 includes computer storage media in the form of volatile and/or nonvolatile memory such as read only memory (ROM) 131 and random access memory (RAM) 132. A basic input/output system 133 (BIOS), containing the basic routines that help to transfer information between elements within computer 102, such as during start-up, is typically stored in ROM 131. RAM 132 typically contains data and/or program modules that are immediately accessible to and/or presently being operated on by processing unit 120. By way of example, and not limitation, Fig. 2 illustrates operating system 134, application programs 135, other program modules 136, and program data 137. As is well known, operating systems provide such low-level functions as module (process and dynamic link library) management, scheduling, interprocess messaging, memory management, file system management, and graphical user interface support.

The computer 102 may also include other removable/non-removable, volatile/nonvolatile computer storage media. By way of example only, Fig. 2 illustrates a hard disk drive 141 that reads from or writes to non-removable, nonvolatile magnetic media, a magnetic disk drive 151 that reads from or writes to a removable, nonvolatile magnetic disk 152, and an optical disk drive 155 that reads from or writes to a removable, nonvolatile optical disk 156 such as a CD ROM or other optical media. Other removable/non-removable, volatile/nonvolatile computer storage media that can be used in the exemplary operating environment include, but are not limited to, magnetic tape cassettes, flash memory cards, digital versatile disks, digital video tape, solid state RAM,

The drives and their associated computer storage media discussed above and illustrated in Fig. 2 provide storage of computer-readable instructions, data structures, program modules, and other data for computer 102. In Fig. 2, for example, hard disk drive 141 is illustrated as storing operating system 144, application programs 145, other program modules 146, and program data 147. Note that these components can either be the same as or different from operating system 134, application programs 135, other program modules 136, and program data 137. Operating system 144, application programs 145, other program modules 146, and program data 147 are given different numbers here to illustrate that, at a minimum, they are different copies. A user may enter commands and information into the computer 102 through input devices such as a keyboard 162 and pointing device 161, commonly referred to as a mouse, trackball, or touch pad. Other input devices (not shown) may include a microphone, joystick, game pad, satellite dish, scanner, or the like. These and other input devices are often connected to the processing unit 120 through a user input interface 160 that is coupled to the system bus, but may be connected by other interface and bus structures, such as a parallel port, game port, or a universal serial bus (USB). A monitor 191 or other type of display device is also connected to the system bus 121 via an interface, such as a video interface 190.

The computer may operate in a networked environment using logical connections to one or more remote computers, such as a remote computer 180. The remote computer 180 may be a personal computer, a server, a router, a network PC, a peer device or other common network node, and typically includes many or all of the elements described above relative to computer 102, although only a memory storage device 181 has been illustrated in Fig. 2. The logical connections depicted in Fig. 2 include a local area network (LAN) 171 and a wide area network (WAN) 173, but may also include other networks. Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets, and the Internet.

When used in a LAN networking environment, the computer 102 is connected to the LAN 171 through a network interface or adapter 170. When used in a WAN networking environment, the computer 102 typically includes a modem 172 or other means for establishing communications over the WAN 173, such as the Internet. The modem 172, which may be internal or external, may be connected to the system bus 121 via the user input interface 160, or other appropriate mechanism. In a networked environment, program modules depicted relative to the computer 102, or portions thereof, may be stored in the remote memory storage device. By way of example, and not limitation, Fig. 2 illustrates remote application programs 185 as residing on memory device 181. It will be appreciated that the network connections shown are exemplary and other means of establishing a communications link between the computers may be used.

## Graphical User Interface

Fig. 3 illustrates a graphical window 300 for displaying and managing objects within multiple hierarchies. An object entry area 302 facilitates user specification of an object that may be part of one or more hierarchies. In this illustration, the object entry area is implemented as a text entry area that allows a user to type in a desired object, such as an object to a person named "John Doe".

A hierarchy display area 304 is used to illustrate the specified object with respect to a selected hierarchy once the user enters a specific object. Here, the depicted hierarchy is a management organizational chart in which John Doe is a member. Hierarchy display area 304 shows nodes 308 with various members in the "Management" hierarchy, including John Doe.

A hierarchy selection area 306 allows a user to select another hierarchy within which the object may be presented. Here, the hierarchy selection area 306 is a drop down menu that illustrates different hierarchies that apply to John Doe, thereby facilitating user selection of a second hierarchy within which to display the specified object, John Doe. In this example, the different hierarchies that apply to John Doe include, management, business units, direct reports, related people, same title, location of office, resources, and e-mail lists. These different hierarchies could be any other type of hierarchy. A system administrator could specify the hierarchies that correspond to objects. Upon selection of another hierarchy from hierarchy selection area 306, the hierarchy display area 304 is changed to illustrate the object with respect to the second intersecting hierarchy.

We now describe a procedure used to display the specified object into window 300. First, a user must specify an object into the object specification control 302. Then, the user selects a "Go" control 310, whereupon the computer

queries one or more databases to determine a list of hierarchies that correspond to the specified object. The data store returns the query results, which include indications of the different hierarchies defined for the specified object.

In the described embodiment, the Extensible Markup Language (XML) is used to format the search query and the search query results between the computer and the data store. XML is a well known document markup language. An example of an XML search query to a data store to determine a list of hierarchies that correspond to the name "John Doe" is illustrated in Table 1.

**Table 1**  
Example of an XML Object Hierarchy Node Search Query

```
Search http://ggr1:81/Post:Query/?Query=<NodeSearch><SearchFor>John Doe
</SearchFor><SearchAttribute>any</SearchAttribute><ElementsOfInterest><Ele
ment>cn</Element><Element>uid</Element>><Element>roomNumber</Elemen
t>><Element>telephoneNumber</Element></ElementsofInterest></NodeSearch>
```

The XML search query illustrates that a node search is being made with respect to John Doe. In this implementation, the result of the node search query is a list of all of the "John Doe's" in the company. Additionally, certain optional elements of interest are included in the query. Such optional elements include user id ("uid"), room number, telephone number, and a common name ("cn"). Table 3 provides an example or a node query, which in this implementation results in returning a hierarchy for "John Doe".



**Table 2**

**Example of a Selected Hierarchy Node Query**

Search [http://ggr1:81/Post:Query/?Query=<NodeQuery><UserID>John Doe</UserID><dimension+up='\\*'>Management</dimension><ElementsOfInterest><Element>cn</Element><Element>uid</Element><Element>title</Element><Element>telephoneNumber</Element><Element> buildingName </Element><Element> roomNumber </Element></ElementsofInterest></NodeSearch>](http://ggr1:81/Post:Query/?Query=<NodeQuery><UserID>John Doe</UserID><dimension+up='*'>Management</dimension><ElementsOfInterest><Element>cn</Element><Element>uid</Element><Element>title</Element><Element>telephoneNumber</Element><Element> buildingName </Element><Element> roomNumber </Element></ElementsofInterest></NodeSearch>)

This query indicates that the specified user ("UserID") is John Doe, and also indicates that the nodes of interest are in the Management hierarchy ("dimension"). More particularly, the query asks for all of those nodes that are parent nodes to John Doe within the Management hierarchy ("dimension+up='\*'"). Optional elements of interest for each identified node are also requested. The values of these optional elements can be used to provide a user with additional information.

Table 3 provides an example of the selected hierarchy node query result.

**Table 3**  
Example of a Selected Hierarchy Node Query Result

```
<Query Results> <Person ID = "19433-4945" Dimension = "Management"> <cn>
Tom Carlson </cn> <uid>TOMC </uid> <TelephoneNumber> +1800555-1212
</TelephoneNumber> <Title> Chairman </Title> <RoomNumber> 2110
</RoomNumber> <Person ID = "19999-4955" Dimension = "Management">
<cn>Jane Jones</cn> <uid>JANEJ</uid> <TelephoneNumber>+1888555-1212
</TelephoneNumber> <Title> Director European Sales</Title>
<RoomNumber>55</RoomNumber> <Person ID = "200009-4005" Dimension =
"Management"> <cn>Carl Thompson</cn> <uid> CARLT </uid>
<TelephoneNumber> +1900555-1212 </TelephoneNumber><Title> Director U.S.
Sales</Title> <RoomNumber>121</RoomNumber> ><Person ID = "33222"
Dimension = "Management"> <cn>Bob Smith </cn> <uid> BOBS </uid>
<TelephoneNumber> +1555555-1212 </TelephoneNumber><Title> Product
Development </Title> <RoomNumber>634</RoomNumber> <Person ID =
"77755-66423" Dimension = "Management"> <cn>John Doe</cn> <uid> JDOE
</uid> <TelephoneNumber> +1777555-1212 </TelephoneNumber> <Title> Sales
and Marketing </Title> <RoomNumber>21</RoomNumber>
```

Objects corresponding to the selected hierarchy ("management") are returned with the query results. Hierarchy display area 304 shows John Doe within a Management Hierarchy.

In this example, as illustrated in Fig. 3, node 308-1 illustrates the specified object – John Doe. Node 308-1 has a parent node 308-2 that illustrates that John Doe's manager, or supervisor is Bob Smith. (For nomenclature purposes, a node having a parent node is called a child node). Similarly, node 308-3 illustrates that Bob Smith's supervisor is Carl Thompson. Node 308-4 illustrates that Carl

1 Thompson's supervisor is Jane Jones. And, node 308-4 illustrates that Jane Jones'  
2 supervisor is Tom Carlson.

3 Although this example, illustrates the specified object within the  
4 Management hierarchy using a tree structure of interconnected parent and child  
5 nodes, a specified object can be represented within a hierarchy using different  
6 graphical objects or even text. For example, a specified object can be illustrated  
7 within a hierarchy using an enumerated list. Moreover, although the nodes are  
8 represented as three-dimensional structures, two-dimensional structures can also  
9 be used.

10 Fig. 4 illustrates the graphical user interface 300 of Fig. 3, but shows the  
11 particular object within a second hierarchy. In this case, the user selects the  
12 "Business Units" hierarchy from the hierarchy selection area 306, and the object  
13 "John Doe" is now dynamically represented within a business units hierarchy. .

14 For example, recall that Fig. 3 illustrated the specified object, " John Doe"  
15 308-1 with respect to a Management hierarchy. Upon user selection of Business  
16 Units hierarchy, as illustrated in hierarchy selection area 306, the graphical user  
17 interface 300 displays the specified object of John Doe within the Business Units  
18 hierarchy. Any of the other hierarchies listed in hierarchy selection area 306 could  
19 also be selected, whereupon the specified object would be illustrated within the  
20 selected hierarchy.

21 The relationship of John Doe within the business unit hierarchy is  
22 illustrated as follows: John Doe 308-1 works directly underneath the sales and  
23 marketing business unit 400-1. The sales and marketing business unit is part of  
24 business divisions 400-2, which in turn is part of product development 400-3.  
25 Product development is part of the European division business unit 400-4, which

1 in turn is part of the U.S. corporation business unit 400-5. In this manner, a user  
2 can select a second hierarchy within which to view a specified object.

### 3 4 **Morphing Between Hierarchies**

5 A structure that illustrates an object within a first hierarchy can be gradually  
6 replaced with a different structure that shows the object within a second hierarchy.  
7 Such gradual replacement is called visual morphing. Morphing from one  
8 hierarchy to another hierarchy provides visual cues to a user that a transition is  
9 being made from viewing an object within a first hierarchy to viewing the object  
10 within a second hierarchy. One technique to morph between first and second  
11 hierarchies is a visual pivot technique.

12 Figs. 5-7 illustrate a visual pivot technique, wherein a first hierarchical  
13 structure is gradually replaced with a second hierarchical structure. Fig. 5  
14 illustrates the hierarchy display area 304 as it would appear immediately after a  
15 user selects to view a specified object, in this example, John Doe within a second  
16 hierarchy of business units. John Doe is represented within the business unit  
17 hierarchy by structure 504. The specified object, John Doe was already  
18 represented within the management hierarchy by structure 502. (See also, Fig. 3).  
19 Node 308-1 is a common pivot point between structures 502 and 504.

20 Fig. 6 illustrates the graphical user interface 300 rotating the business unit  
21 structure 504 around the common pivot point 308-1 toward the management  
22 structure 502. Fig. 7 further illustrates the graphical user interface 300 rotating the  
23 business unit structure 504 around pivot point 308-1 closer yet to the management  
24 structure 502. This rotational pivot will continue until structure 504 is rotated onto  
25 the structure 502, whereupon the structure 502 will be faded out.

Fig. 4 represents the fully rotated business unit structure 504, the structure having been rotated onto the management structure 502. The management structure 502 has been faded out, leaving only the business unit structure 504. In this manner, the second structure was visually pivoted around a pivot point that was common to both structures, and the first hierarchical structure was replaced by the second hierarchical structure.

The visual pivot technique can be modified. Consider that the above example illustrated a pivot about a vertical axis through the selected node with the original hierarchy remaining in place as the second hierarchy was pivoted onto it – appearing somewhat like a book turning a page. The visual pivot, however, could also be performed by moving both hierarchies about the axis until the second hierarchy is in the previous position of the original hierarchy, whereupon the original hierarchy is faded away.

The visual pivot could be performed with one hierarchy being offset vertically to one-another, after the pivot being complete, the second hierarchy being shifted to remove the offset onto the original hierarchy, whereupon the original hierarchy is faded away. Different axis as well as different pivot speeds can be used. Moreover, there are other procedures, including other morphing procedures that can be used to gradually replace a structure that represents an object within a first hierarchy with a different structure that shows the object within a second hierarchy.

Fig. 8, for example, illustrates another visual morphing technique, a visual slide technique, wherein a first hierarchical structure is gradually replaced with a second hierarchical structure. Hierarchy display area 304 appears as it would immediately after a user selects to view the specified object, in this example, John

1 Doe within a second hierarchy of business units. John Doe was already  
 2 represented within the management hierarchy by structure 800. (See also, Fig. 3).  
 3 John Doe within the business unit hierarchy is represented by structure 802.  
 4 Notice that node 308-1, which represents the specified object of John Doe, appears  
 5 in both structures.

6 To gradually replace structure 800 with structure 802, structure 802 is  
 7 moved laterally until it overlies structure 800. This lateral movement is denoted  
 8 by directional arrow 804. At this point, structure 800 is faded out, leaving only  
 9 structure 802. Fig. 4 represents the final result of structure 802 having been  
 10 laterally moved onto structure 800, whereupon structure 800 was faded out,  
 11 leaving only structure 802. In this manner, a first hierarchical structure is  
 12 gradually replaced with a second hierarchical structure.

13 In another example (not illustrated with a figure), the particular object is  
 14 shown within each hierarchy of objects on an index card-like user interface control  
 15 (hereinafter "control"). To switch from one hierarchy to another, the user drags an  
 16 object from a hierarchy to a rolodex-like control, whereupon the dragged object is  
 17 illustrated within multiple other hierarchies on superimposed, or laterally  
 18 positioned index card-like controls. The user then selects the index card-like  
 19 control that shows the dragged object within the particular hierarchy of interest,  
 20 whereupon the selected index card-like control is enlarged to illustrate the  
 21 particular object within the selected hierarchy. In this manner, a first hierarchical  
 22 structure is gradually replaced with a second hierarchical structure.

23 In another example (not illustrated with a figure), a card-like structure that  
 24 illustrates a particular object within a second hierarchy is unfolded onto an  
 25 illustration of the particular object within a first hierarchy. To accomplish this, the

1 card like structure is first shown superimposed on the particular object within the  
2 first hierarchy – the superimposed card is folded and partially open, such that at  
3 least a portion of the second hierarchy is shown at the same time that the first  
4 hierarchy is visible. The card is folded at an axis defined by the particular object.

5 To morph the second hierarchy onto the first hierarchy, the card illustrating  
6 the particular object within the second hierarchy is gradually unfolded onto the  
7 first hierarchy until the card is completely open – whereupon the illustration of the  
8 first hierarchy is completely faded away. In this manner, a first hierarchical  
9 structure is gradually replaced with a second hierarchical structure. Thus, different  
10 techniques can be used to replace an illustration showing a particular object within  
11 a first hierarchy with another illustration showing the particular object within a  
12 second hierarchy.

### 13 **Displaying Multiple Specified Objects within a Hierarchy**

14  
15 Fig. 9. illustrates a graphical user interface for displaying and managing  
16 data within multiple intersecting hierarchies of information. In particular, Fig. 9  
17 illustrates the that graphical user interface 300 further comprises a search results  
18 area 900 to display one or more objects that correspond to a search for an object  
19 specified in the object entry area 302. Here, a user is looking for “John Doe”, and  
20 the search identifies several different John Does.

21 Search results area 900 also facilitates user selection of an additional object  
22 902. Upon selection of an object identifier 902 of an additional object 902, the  
23 additional object replaces any details in hierarchy display area 304 and displays  
24 the additional object within the current hierarchy. Upon user selection of a “>>”  
25 control 904, a corresponding additional object is added to the details displayed in

1 hierarchy display area 304, such that the additional object is displayed with other  
2 specified objects in the current hierarchy. The object identified 902-4 is  
3 highlighted to illustrate that "John Doe" is illustrated within the current hierarchy.

4 Fig. 10 illustrates that upon selection of a different object identifier 902 of  
5 (this is indicated by the selection arrow and the bolded font), the selected  
6 additional object replaces any details in hierarchy display area 304, whereupon the  
7 selected additional object is displayed within the current hierarchy. Notice that  
8 nodes 308, representing John Doe 902-4 within the management hierarchy (see  
9 Fig. 9) were replaced with nodes 1000, representing John F. Doe within the  
10 management hierarchy.

11 Fig. 11 illustrates specification and display of multiple objects with respect  
12 to a hierarchy. Upon user selection of a ">>" control 904, the control's  
13 corresponding object 902 is also displayed within the current hierarchy without  
14 replacing any other details in hierarchy display area 304.

15 For example, upon selection of control 904-4 corresponding to John Doe  
16 902-4, both John Doe and John F. Doe are displayed within the current hierarchy  
17 of information, the management hierarchy. John Doe is represented within the  
18 management hierarchy with nodes 308. John F. Doe is represented within the  
19 management hierarchy with nodes 1000. This example, further illustrates that  
20 John Doe and John F. Doe share a management entity, represented by node 308-4  
21 Jane Jones. In this manner, complex relationships between objects can be  
22 visualized within a hierarchy of information.

23 In this implementation, if a new portion of the display would replace part of  
24 the current display, as opposed to simply adding it, the part of the display being  
25 replaced is moved by animation to make room for the new information.



Moreover, when removing an item from the display from a list of several items, parts of the display may be moved in an animation before fading out the item to be removed. The part of the display that is moved can correspond either to the item to be removed or the items that are not being removed

As discussed above, the current hierarchy can be changed by selecting a different hierarchy from hierarchy selection area 306, whereupon the specified objects 308-1 and 1000-1 will be displayed within the different hierarchy. Additionally, using procedures discussed above, the structures representing the specified objects within a first hierarchy can be visually morphed into different structures that represent the specified objects in a different hierarchy.

Graphical user interface 300 further includes a selected object display area 1100 to display a list that includes an indication of the object specified in object entry area 302 along with one or more indications of the additional objects selected from search results area 900. The selected object display area 1100 facilitates the removal of a listed object from hierarchy display area 304 so that, upon selection of a listed object, the selected object is removed from hierarchy display area 304.

For example, hierarchy display area 304 illustrates both "John Doe" and "John F. Doe" within a management hierarchy. Selected object display area 1100 indicates that objects "John Doe" and "John F. Doe" are illustrated with respect to the current hierarchy. Upon selecting "John F. Doe" from selected object display area 1100 and upon further selecting control 1102, the information corresponding to "John F. Doe" (nodes 906) is removed from hierarchy display area 304. Moreover, the indication of "John F. Does" is removed from selected object

1 display area 1100. In this manner, a user can remove information from hierarchy  
2 display area 304.

3 Fig. 12 illustrates that only information corresponding to John Doe is  
4 shown in hierarchy display area 304 upon user selection of John F. Doe from  
5 selected object display area 1100 and control 1102.

6 Graphical user interface 300 further includes a filter selection area 1200 to  
7 provide a filter to further determine the object identifiers 902, or additional objects  
8 to be displayed in search results area 900. The filter(s) corresponds to the object  
9 specified in object entry area 302. In this example, because the specified object is  
10 a name, the filter "any part of the name" corresponds to a name. The object  
11 identifiers 904 are those objects that matched either "John" or "Doe". Moreover,  
12 if the filter were changed to one of "full name", "first name", "last name", "e-mail  
13 alias", "title", or the like, the additional objects would be those objects that met the  
14 criteria of the selected filter.

15 A node 308 displayed within a hierarchy includes context sensitive  
16 information 1202. Upon user selection of a node (e.g., with a right-mouse button  
17 click) the context sensitive information is displayed to provide indications of the  
18 hierarchies that correspond to the node. Upon user selection of one of the node's  
19 displayed hierarchies, the node is displayed with respect to the selected hierarchy.

20 Fig. 13 is a flowchart that illustrates a procedure 1300 for displaying  
21 objects within multiple hierarchies of information. At step 1302, the procedure  
22 specifies an object having first and second intersecting hierarchies. At step 1304,  
23 the procedure illustrates the object with respect to the first hierarchy. At step  
24 1306, the procedure presents the second intersecting hierarchy for selection to  
25 illustrate the object with respect to the second intersecting hierarchy. At step

1 1308, the procedure selects the second intersecting hierarchy. At step 1310, the  
2 procedure, in response to the selecting, illustrates the object with respect to the  
3 second intersecting hierarchy. In this manner, objects are displayed within  
4 multiple intersecting hierarchies of information.

### 5 6 **Managing Objects within Hierarchies and Polyarchies**

7 An object that is displayed with a hierarchy can be managed by creating  
8 objects in a hierarchy, deleting an object from a hierarchy, or changing the object's  
9 relationships with respect to other objects in the hierarchy.

10 For example, to add an object to a hierarchy, an insert command (not  
11 shown) can be selected, whereupon a new object would be displayed in the  
12 hierarchy with a type in box for typing in an object identifier. To remove an object  
13 from a hierarchy, an object can be selected and a deleted command can be issued,  
14 whereupon the object will be removed from the hierarchy. In both of these  
15 instances, corresponding databases would be updated accordingly to reflect the  
16 change in the hierarchy.

17 In yet another example, referring to Fig. 3, if John Doe 308-1 is promoted  
18 to Carl Thompson's 308-3 manager, node 308-1 is selected and dragged to the  
19 location between nodes 308-3 and 308-4, whereupon node 308-1 is dropped. As a  
20 result, John Doe will be represented within the management hierarchy as both  
21 manager to Carl Thompson and Bob Smith – corresponding databases are updated  
22 to reflect the management change. Drag and drop techniques are well-known. In  
23 this manner, relationships between objects within an intersecting hierarchy can be  
24 managed.

1 **Conclusion**

2       Although details of specific implementations and embodiments are  
3 described above, such details are intended to satisfy statutory disclosure  
4 obligations rather than to limit the scope of the following claims. Thus, the  
5 invention as defined by the claims is not limited to the specific features described  
6 above. Rather, the invention is claimed in any of its forms or modifications that  
7 fall within the proper scope of the appended claims, appropriately interpreted in  
8 accordance with the doctrine of equivalents.

9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25